Today

- BPF
- intro to security
- denial of service attacks

raw and data-link

- raw sockets allow you to read all packets for your host
- can write your own IP header with raw sockets
- data-link layer can be accessed via most OS's
- *libpcap* is nice wrapper
- with *libpcap* can read and write Ether frames
- enable promiscuous mode and read all packets on the wire
- what tcpdump uses

raw sockets

- need privileges (root)
- application can read protos icmp, igmp packets (any with IPPROTO_RAW)
- can read/write IPv4 datagrams with a proto field not supported by the kernel
- with IP_HDRINCL socket option, can build IP header! popular with hackers. kernel calculates checksum
- on read, don't get UDP/TCP, but most ICMP, all IGMP, all IP datagrams with unsupported proto fields – watch out, you'll get more packets than you expect
- used by ping, traceroute, mrouted, hackers

```
sd=socket(AF_INET, SOCK_RAW,
IPPROTO_ICMP);
```


data link layer

- normally network interface receives only packets destined for host (or broadcast or multicast)
- most Ether, FDDI, and token ring interface cards allow promiscuous mode to be enabled
- most UNIX systems have software interface to access data-link layer packets (BPF, NID, DLPI, SOCK_PACKET)
- requires root privileges
- *libpcap* provides wrapper
- packages are availabe for windows/NT too
- with data link access, one can develop protocol analyzers, network analyzers, intrusion detectors, sniffers

BPF

- data-link layer access for BSD, Dec Unix
- kernel-level packet filtering
- lower overhead than application filtering
- usually just header bytes passed

see McCane paper

libpcap

get it and tcpdump at ftp.ee.lbl.gov

- implementation-independent data-link layer access
- used by tcpdump
- can read/write interface or file
- filter language
  "host bob and (port 23 or port 21)"
- buffering
- but YOU have to decode the packets
- often you can do it with tcp-dump and perl/awk

example

```
  we want to print time and seq # for a TCP session
  argv[1] will be  "tcp and dst host bob and port 2000"
    ....
 main(argc,argv){
  device = pcap_lookupdev(errbuf);
  pd = pcap_open_live(device, snaplen, 1, 1000, errbuf);
  pcap_lookupnet(device,&net,&netmask,errbuf);
  pcap_compile(pd,&code,argv[1],1,netmask);
  pcap_setfilter(pd,&code);
  pcap_loop(pd, 0, read_pkt, NULL);
}
```



```
    * each packet has the following header
    struct pcap_pkthdr {
        struct timeval ts;      /* time stamp */
        bpf_u_int32 caplen;     /* length of portion present */
        bpf_u_int32 len;        /* length this packet (off wire) */
    };
  ---------

void
read_pkt(u, h, p)
        u_char *u;  /* pointer from pcap_loop */
        const struct pcap_pkthdr *h;  /* header */
        const u_char *p;    /* data-link packet */
{
   pktsec =h->ts.tv_sec;
   ts = h->ts.tv_sec + 1.e-6*h->ts.tv_usec;
   if (ts0 == 0) ts0=ts;
   ts = ts-ts0;
   ...  chop off data-link header (varies by media: pcap_datalink())
   ... align data (p) on IP struct boundary (ipbuf)
   packet_ip(ipbuf);
}
```

```
packet_ip(ip)
register struct ip *ip;
{
   struct tcphdr *tp;
   tp = (struct tcphdr *)( (char *)ip + 4*ip->ip_hl);
   if (ip->ip_p == IPPROTO_TCP ){
        seq = ntohl(tp->th_seq);
        if (seq0 == 0) seq0 = seq;
        seq = seq-seq0;
        printf("%f %u\n",ts,seq);
    }
 }
```


network security

Objective: scare you

- is there a problem?
- network threats
- network defenses

Computer security

Assets

- hardware
- software
- information

Goals

- integrity
- privacy
- availability

Threats

- interruption
- interception
- modification
- fabrication

virus, worm, backdoor, sniffer, spoof, hijack, denial of service

Countermeasures/Defenses

- education
- policy and procedures
- physical protection
- authentication
- authorization
- auditing (intrusion detection)
- encryption

Threats/countermeasures – a never ending cycle ...

Are we at risk?

assets

air defense	nuclear weapons systems
command and control	Taco Bell
banking	electronic funds transfer
power grid	air traffic control
phone system	elevators
traffic signals	trains
corporate email	grades
refinery	stock exchange
DMV	TV/radio
medical records	police records
personnel records	payroll

Information warfare

Point, click, attack

- cookbook attacks
- your little brother could do it
- underground resources
- tools designed by troubled genious
  - deep understanding of OS (source files)
  - look for known vulnerabilities (overflows)
  - lots of time
  - adaptable, avoids countermeasures

The Attackers

- amateur
- insider (greed, disgruntled)
- kids
- hackers
- criminals
- spies
- sociopath (terrorist/vandal)

Why?

- money
- retribution
- sport
- pathological
- political/military

easy to do, hard to catch, harder to prosecute

victimless crime? just a prank?

Ancient History

" A long time ago in a galaxy far, far way ..."

1500 BC first written cryptography

500 BC Hebrew substitution cipher

50 BC Caesar cipher

1876 telephone invented

1878 first report of teenagers kicked off phone system for pranks

1917 one-time pad

1923 Enigma machine

1948 Captain Midnight decoder ring (Ovaltine)

1950's/60's single user then batch computing

recent history

'64 teletype/acoustic coupler

'67 DEC10 timesharing

'69 ARPANET (email)

'70 DEC 11 / UNIX

'71 Captain Crunch – 2600

'74 DES

'75 crypt for passwd

'76 public key crypto / Ethernet

'77 Apple II / uucp/USENET

'79 VAX / BSD UNIX (free)

'80 DECnet / MFEnet / SNA / CSnet / BITNET / MS DOS

'81 Mitnick (17) steals Pac Bell manuals

'84 ORNL on Internet (ARPAnet/MILNET)

'85 Sun workstations (sniffers)

'86 first virus/ LBL cuckoo's egg

'88 Morris worm (hit ORNL)

'91 PGP

'93 Mosaic/www point/click/attack

'94 ORNL/MSR breakin

'94 Linux (free) / rootkit

'95 Mitnick attack SDSC / SATAN / SSL

'98 smurf attack

'99 AES

network attacks

- denial of service (DoS)
- impersonation/spoofing
- account capture
- scan and exploit
- advanced

denial of service

- overload service/net
- crash server or your machine
- usually done with bogus source IP address(es)
- difficult to block/filter (2nd order denial of service)
- difficult to trace (open research)

service overload

- flood mail server (anonymous mail)
- flood DNS server, syslogd (UDP)
- flood log files (syslogd, detectors)
- SYN flooding
- UDP with spoofed source address, source port 7 (echo), legit destination, destination port 7
- or destination local broadcast address !!

routers can/should prevent local addresses from arriving from an external interface and should not pass in packets with broadcast destinations.

smurf attack

- hacker on his slow dial up connection, sends ICMP echo with broadcast destination (preferably of a net with high speed link)
- source address is spoofed and the target of the flood of ICMP replies from the destination net
- if the target net has a slow link, then whole target subnet may be slowed
- hackers like these high-leverage attacks: they send one packet and generate lots of nasty traffic
- hackers also use broadcast ICMP echo (with a legit source address) to try and map active hosts on a destination net

IP implemenations

bugs

- OS hang or crash (blue screen)
- land attack – SYN packet with both source and destination address the target machine
- teardrop attack – overlapping or bogus offsets in IP fragments
- mangled IP options, TCP options, bogus TCP flag combinations, etc.

patch your OS

other

- bogus "ICMP host unreachables" sent to target
- insider attack: "ICMP redirects" to route traffic to hacker's bogus router or to blackhole
- insider attack: bogus routing protocol updates (RIP)
- may be able introduce bogus routes into Internet routers

next time

IP security – more attacks